

Stored Procedures/Functions - Parameters (PostgreSQL Version)

Parameters in a PostgreSQL stored procedure (or function) allow values to be passed in and used inside the procedure (or function). These parameters can be used for filtering data, performing calculations, or controlling logic within SQL statements.

In PostgreSQL, parameters are declared inside parentheses after the procedure name.

Each parameter must have:

- A **name** (which starts with a colon or nothing at all — not @ as in SQL Server),
- A **data type**, and
- An optional **default value** can be assigned (for example, Default Null or = Null), which is used if no value is provided when calling the procedure or function.
- To test whether a parameter was passed, use Is Null, not = Null.

Example: Procedure with a Parameter

The following example creates a stored procedure that accepts a Student_ID parameter:

```
Create Or Replace Procedure PR_LookupStudent
```

```
(p_StudentID integer = Null)
```

```
Language plpgsql
```

```
As $body$
```

```
Begin
```

```
    If p_StudentID Is Null Then
```

```
        Raise Notice 'StudentID was not passed in';
```

```
    Else
```

```
        Raise Notice 'Looking up student with Id: %', p_StudentID;
```

```
    End If;
```

```
End;
```

```
$body$;
```

Executing a Stored Procedure with a Parameter

You can execute a stored procedure using the CALL statement:

Call PR_LookupStudent(12345);

Or, if you want to use the default parameter value: Call PR_LookupStudent ();

Example: Procedure with multiple Parameters

Parameters are passed in order, separated by commas.

Create Procedure PR_AddStudent

(p_StudentID integer, p_FirstName varchar, p_LastName varchar,
p_Gender char, p_BirthDate date)

Language plpgsql

As \$body\$

Begin

Insert Into student

(StudentID, FirstName, LastName, Gender, BirthDate)

Values

(p_StudentID, p_FirstName, p_LastName, p_Gender, p_BirthDate);

End;

\$body\$;

Call the procedure with multiple parameters

Call PR_AddStudent (100, 'Jane', 'Doe', 'F', '01-Jan-2000');

Example: Function with a Parameter

Here's a stored function that retrieves a student's full name based on a passed in StudentID. If no value is passed in, it returns a message instead of the student's name.

```
Create Function FN_Get_Student_Name
    (p_StudentID integer = Null)
    Returns varchar
Language plpgsql
As $body$
Declare
    v_StudentName varchar(80);
Begin
    If p_StudentID Is Null Then
        Return 'StudentID was not passed in';
    Else
        Select          FirstName || ' ' || LastName
        Into             v_StudentName
        From             Student
        Where             StudentID = p_StudentID;
        Return v_StudentName;
    End If;
End;
$body$;
```

Run the function with a Parameter (or not)

```
Select FN_Get_Student_Name(199899200);
-- or --
Select FN_Get_Student_Name();
```